IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants:	Kevin M. Conley et al.		
Title:	Management Of Non-Volatile Memory Systems Having Large Erase Blocks		
Application No.:	10/749,831	Filing Date:	December 30, 2003
Examiner:	Eland, Shawn	Group Art Unit:	2188
Docket No.:	0084567-247US0	Conf. No.:	9380

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Further to the Notice of Appeal filed in the above-identified patent application on April 20, 2009, this Appeal Brief is being submitted to the Board of Patent Appeals and Interferences under the provisions of 37 C.F.R. § 41.37 that became effective September 13, 2004.

TABLE OF CONTENTS

I. REAL PARTY IN INTEREST

The real party in interest is SanDisk Corporation, a Delaware corporation, the assignee of the entire right, title and interest in the present application from the applicants Kevin M. Conley and Carlos J. Gonzalez, by way of a written Assignment dated December 24, 2003 and recorded in the United States Patent and Trademark Office at reel/frame 014613/0295 (2 pages) on May 10, 2004.

II. RELATED APPEALS AND INTERFERENCES

Based upon information and belief, there are no appeals or interferences that could directly affect or be directly affected by or have a bearing on the decision by the Board of Patent Appeals and Interferences in the pending appeal.

III. STATUS OF CLAIMS

The final rejection of claims 2-4 and 26-41 is being appealed, all of the claims that remain pending in the present application. These claims are reproduced in the Claims Appendix hereto.

The present application was originally filed with claims 1-25. In response to a Requirement for Restriction, claims 5-25 were withdrawn from consideration and then cancelled. Original claim 1 was also cancelled, original claims 2-4 amended and new claims 26-41 added during the course of prosecution.

IV. STATUS OF AMENDMENTS

An Amendment after Final Office Action was filed April 20, 2009, concurrently with the Notice of Appeal herein. This Amendment cancelled a term from each of the independent claims 2, 28, 31, 36 and 40 that caused the Examiner to reject these claims under 35 U.S.C. § 112, second paragraph, in order to remove the indefiniteness rejection from those necessary to take on appeal. By an Advisory Action mailed May 5, 2009, it was stated that the amendment was entered for the purpose of appeal and overcame the rejection under 35 U.S.C. § 112, second paragraph.

V. SUMMARY OF CLAIMED SUBJECT MATTER

All of the claims pending in the present application define a method of operation of a non-volatile memory system having blocks of memory cells. As background, Figure 3 of the present application shows individual planes (sub-arrays) 0 and 1 in a memory system. Each of the planes has its memory cells divided into sixteen such blocks 0-15. The blocks are in turn individually divided into a number of pages of memory cells, sixteen pages 0-15 being shown in the block illustrated in Figure 4. The memory cells of a block are simultaneously erased (the unit of erase; see application ¶0008 & ¶0045) before new data are written into it. Data are written (programmed) into a block in units of the page (a programmable unit; see application ¶0008 & ¶0046). As shown in the address tables of Figures 7A and 7B, the blocks of memory cells are identified by physical block numbers (PBNs), and blocks of data programmed into the designated physical blocks are identified by logical block numbers (LBNs). Pages of memory cells within physical blocks are identified by their offset positions within their blocks. (See, for example, page 0, page 1, etc. of the block of Figure 4.) Each page contains one or more units (such as sectors) of data received from a host (see application ¶0008 & ¶0046, and the example data sector shown in Figure 5).

When a large number of units of data with sequential logical addresses relative to the storage capacity of a memory cell block are received by the memory system for programming, it is efficient to program that data in an erased block of memory cells in sequentially addressed pages. But when only one or a few host units of data have sequential logical addresses relative to the capacity of a memory cell block, it can be inefficient to program that small amount of data in an erased block, particularly when the received data have the same logical addresses as data already programmed and are therefore updating the original data. Therefore, data are programmed differently depending on the number of units of data having sequential logical addresses relative to the data storage capacity of a block of memory cells. (See the Summary of the Invention section of the present application, ¶¶0015-0016.)

An example of programming a relatively small amount of data is illustrated in application Figure 9. (Described in application ¶¶0056-0058.) Each of the independent claims 2, 28, 31, 36 and 40 on appeal call for "designating" or "allocating" one of the memory cell blocks for programming relatively small amounts of data having sequential logical addresses. Such a block is identified in the example of Figure 9 as the E1 block. Figure 9 shows an updated version of

data stored in pages 7-10 of original block 3 being programmed into the next erased pages in order of the E1 block. It was chosen to program this amount of data into the designated E1 block, instead of some other block, because the number of pages of data falls below a pre-set proportion of the total number of pages within a block. Examples of this pre-set proportion include 50% and 75% (see application ¶¶0064-0066). In the example of Figure 9, the memory cell blocks have 16 pages of memory cells that can therefore store up to 16 pages of data, and 4 pages of updated data are being programmed. These 4 pages of data are programmed into the designated E1 block because their number is less than a pre-set proportion of 8 pages, using the 50% number given as an example in application ¶0065.

The pre-set proportion is specified by dependent claims 26, 27, 30, 33, 35, 37, 39 and 41 to be within a range of 25-75 percent of the storage capacity of a memory cell block. This is described in ¶0065 of the present application. Some of the appealed claims additionally call for programming larger numbers of units of data with sequential logical addresses into another designated block, called the E2 block in the present application. (See application ¶0015 & ¶0056, for example.)

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The Board is being asked to review the following grounds of rejection:

(a) Claims 2-4, 28, 29, 31, 32, 34, 36, 38 and 40 under 35 U.S.C. § 102(b) over patent application publication no. US 2002/0099904 A1 of Conley (hereinafter "Conley"), which includes all of the independent claims 2, 28, 31, 36 and 40 on appeal; and

(b) Dependent claims 26, 27, 30, 33, 35, 37, 39 and 41 under 35 U.S.C. § 103(a) over Conley and patent application publication no. US 2002/0034105 A1 of Kulkarni et al. (hereinafter "Kulkarni").

VII. ARGUMENT

Rejection under 35 U.S.C. § 102(b)

The primary issue presented by this appeal is whether the cited Conley reference discloses *designating* or *allocating* a memory cell block for programming a number of units of data having sequential logical addresses that is less than a *pre-set proportion* or *pre-set fraction*

of the given number of units of data that may be stored in a memory cell block. (The words in italics are from the appealed independent claims.)

It is agreed that Conley describes programming a number of units of data having sequential logical addresses in any one of a number of memory cell blocks that has enough erased capacity to store these units of data. But Conley does not designate or allocate one of the blocks for programming a number of units of data with sequential logical addresses that is less than a pre-set proportion or fraction of the given total number of units of data that a block may store. The claimed criterion for selecting a block in which received data are programmed is much different than Conley's criterion.

With reference to Figure 14 and paragraph 0062 of Conley, which are repetitively referenced in the final Office Action, the programming of an amount of data in any memory cell block that has the capacity to store that amount is asserted by the final rejection to anticipate the appealed independent claims. (See the bolded portions of the Office Action on pages 4-7, 9, 11 and 12.) It is clear from Conley's operational flowchart of Figure 14 that a block is selected for the storage of a particular amount of data received from a host by comparing that amount with the storage space available in various blocks. It if is determined in step 53 that there is enough data to fill a block, the data are programmed in steps 55 and 57 into a fully erased block. But if it is determined in step 53 that there is less than a full block's worth of data to be stored, the data are written in a partially written block in step 63 if one with enough available erased capacity is available. If a partially written block with enough erased capacity is not available, then the data are written in step 65 into a fully erased block.

There does not appear to be any disagreement about this aspect of Conley. Where there is disagreement is whether this discloses *designating* or *allocating* a memory cell block for programming a number of units of data having sequential logical addresses that is less than a *pre-set proportion* or *pre-set fraction* of the given number of units of data that may be stored in a memory cell block. The Examiner concludes it does by interpreting the "pre-set proportion" and "pre-set fraction" terms of the claims as including 100% of the given number of units of data that may be stored in a memory cell block. Any amount of data stored in a block is therefore taken to fall within the scope of the claims, including an amount of data that fills 100% of the capacity of any block.

In this regard, the Office Action (p. 14, lns. 4-7) states as follows:

> A proportion or a fraction may be defined as a small part of something, such as ¼ of element X. But proportion also can be defined as a ratio, such as 1 out of 1, and a fraction as merely one number divided by another, such as 1/1. Both proportion and fraction in the previous examples means 100% so the Examiner's interpretation still stands. Furthermore, fractions and proportions could also mean more than one, as in 2/1, 3/2, 4/1, etc.

As a result of this interpretation of the claims, the examiner finds Conley's programming of any number of units of data into any block having enough erased capacity to anticipate the appealed independent claims and others. With this erroneous interpretation of the claims and by also ignoring that data are claimed to be written into a designated or allocated memory cell block, the Examiner finds Conley to anticipate the claims.

The Examiner's above-quoted interpretation of "proportion" is contrary to the way the term is used in the present application. For example, a "large proportion of the pages of a block are being updated, . . " (present application paragraph 0064, 2nd sentence) is referring to where "the number of pages being updated . . is small in comparison with the total number of pages in the individual block" of the preceding sentence. In paragraph 0065, an example of where a "proportion is equal to or greater than 50 per-cent . ." is given. The term "proportion" is clearly being used to refer to a number of pages less than the total number of pages in a block. This is contrary to the Examiner's interpretation of "proportion" of the number of pages in the block as any number including 100% or more.

Furthermore, in accordance with this application disclosure, claim 2 was amended to add "the pre-set proportion being less than the given number" of units of data stored by a block. This makes it absolutely clear that the criterion of a "pre-set proportion" of the given number of units of data that are stored by a block is something less than 100% of the given number. How the Examiner can still interpret this limitation of claim 2 to cover 100% or more of the pages of a block is certainly not understood.

With the claim criterion of a "pre-set proportion" and "pre-set fraction" of the given number of units of data stored in a block properly interpreted to be something less than 100% of the given number, Conley cannot anticipate the method of the independent claims. In accordance with the flow chart of Figure 14 of Conley discussed above and relied upon by the Examiner, the criterion for whether any particular one of the blocks of memory cells will be

chosen to store a certain amount of data is the amount of available erased capacity in the particular block. If there is enough capacity in a particular block of memory cells for the received amount of data, then it will be programmed in that block. The claimed criterion as to whether a received number of units of data with sequential logical addresses is less than a pre-set proportion or fraction of the capacity of a designated or allocated block is not suggested by Conley's different criterion that applies to all the blocks of the memory cell.

In addition to the novelty discussed above for the independent claims over Conley, independent claims 2 and 28 recite that the number of units of data with sequential logical addresses that are compared with the "pre-set proportion" are received by the memory system with a plurality of successive host commands. That is, it is the number of units of data received with successive commands that is compared with the number that is the pre-set proportion of the storage capacity of a block when making the decision whether those data are to be written into the designated block or not. Multiple units of data with sequential logical addresses are most typically received with a single program command but claims 2 and 28 call for applying the same criterion for use of the designated block when the data with sequential logical addresses are received with a plurality of programming commands. This is additional point of novelty over Conley.

Therefore, for these reasons, each of the independent claims 2, 28, 31, 36 and 40 are submitted to have been erroneously rejected as anticipated by Conley. These claims are submitted to be novel and patentable over Conley. All of the remaining appealed claims are therefore also submitted to be allowable since they are in dependent form.

Rejection under 35 U.S.C. § 103(a)

There is the further issue of the rejection for obviousness of dependent claims 26, 27, 30, 33, 35, 37, 39 and 41 over a combination of Conley and Kulkarni. Kulkarni describes filling 50% of a buffer RAM memory block with a small amount of the data of a very large file before that data are transferred from the RAM to non-volatile memory. Rather than requiring a 10 Mb RAM for use in creating a 10 Mb image file in non-volatile memory, one of its examples, one or more RAM blocks that individually have a capacity equal to that of a 64 Kb non-volatile memory block size are provided. After one-half of a RAM block is filled with the portion of the large data file, that data are then written to the flash memory. The rejection is understood to be

based on the alleged obviousness of imposing a data storage limit of 50% of their capacity on some of Conley's blocks.

Based on this understanding, such a combination of Conley and Kulkarni would not meet the terms of the claims. The appealed claims do not impose a limit on how much data an individual block may hold. Rather, the criterion of the appealed claims is whether the number of units of data having sequential logical addresses less than a fixed number, namely a pre-set proportion or fraction of the data storage capacity of the individual blocks. If less than this proportion, the data are written to a block designated or allocated to receive such small data writes. This improves the performance of the memory system, primarily by reducing the need for garbage collection.

In addition, it is not seen that Kulkarni describes filling up non-volatile memory blocks as defined in the present application claims with only 50% of their capacity, a fact alleged as a basis for the obviousness rejection. Kulkarni's memory block is defined to "represent a minimum size of data that can be written to the flash memory 132 during a write procedure." (Kulkarni, ¶ 0034, lines 6-8; see also ¶ 0012, lines 1-3, and ¶ 0015, line 3.) However, the block is defined in the present claims as the number of cells that are simultaneously erasable. Kulkarni's block is similar to the page defined in the present application as a unit of programming (see ¶ 0008 of the present application), with many pages included in each of the present application's block. A scan of the text of Kulkarni reveals that the word "erase" is not even present, so we do not know anything about Kulkarni's unit of simultaneous erase. Kulkarni's block of memory cells is a unit of programming, while the block of the present application claims is a unit of erase. These are not at all comparable. It is because a complete block of memory cells is erased before data are written into even a small portion of the block that creates the challenge in managing operation of flash memory that is met by the present invention. Kulkarni is not even remotely concerned with such a challenge.

Also, it is not clear that Kulkarni discloses filling up its non-volatile memory blocks to only 50% of their capacity. Data are written to a non-volatile memory block from an equal sized buffer RAM memory block when the RAM block is 50% full but this does not compel the conclusion that only 50% of the non-volatile memory block is used. This would certainly be wasteful of non-volatile memory capacity, resulting in not using about one-half of the memory. Indeed, writing into only 50% of a non-volatile block violates Kulkarni's constraint that "The

memory blocks 140 represent a minimum size of data that can be written to the flash memory 132 during a write procedure." (Kulkarni, ¶ 0034, lines 6-8.) Nor is it clear that only 50% of the RAM block is used, since Kulkarni only states that the 50% point is when data are then written from the RAM into the non-volatile memory. One way to reconcile this apparent contradiction is to consider Kulkarni as beginning the transfer of data from the RAM block to the non-volatile memory block when the RAM block becomes about 50% full, with this transfer continuing as the RAM block continues to receive data. Kulkarni is certainly not clear on how its flash memory blocks are used.

In any event, Conley and Kulkarni describe completely different techniques for operating non-volatile memory that address different problems to be overcome. Conley's goal is to improve memory performance by avoiding copying unchanged data from one block to another, and to avoid the need to change a flag or other data in the pages of a block containing the original unchanged data. (See Conley, ¶ 0008, for example.) Kulkarni's goal, on the other hand, is to minimize the size of buffer RAM memory that is necessary to store large amounts of data of an image. (see Kulkarni, ¶ 0013, for example.) These different goals are reached by different memory system operations. It is therefore not understood how one of ordinary skill would have found it obvious to combine certain unrelated features selected by the Examiner.

Conclusion

For the reasons given above, it is respectfully submitted that each of the appealed claims is patentable over the cited references. A decision of the Board reversing the final rejection of the Examiner, with instructions that the present application be allowed, is solicited.

FILED VIA EFS

Respectfully submitted,

June 22, 2009

Gerald P. Parsons
Reg. No. 24,486

Date

Davis Wright Tremaine LLP
505 Montgomery Street, Suite 800
San Francisco, CA 94111-6533
(415) 276-6500 (main)
(415) 276-6534 (direct)
(415) 276-6599 (fax)
Email: geraldparsons@dwt.com

VIII. CLAIMS APPENDIX

The following is the text of the appealed claims:

2. A method of writing data into a non-volatile memory system of a type having blocks of memory cells that are simultaneously erasable and which individually store a given number of host units of data, comprising:

designating a first of the blocks for storage of a number of units of data with sequential logical addresses that is less than a pre-set proportion of the given number, the pre-set proportion being less than the given number,

thereafter responding to a plurality of successive host commands to write a number of units of data less than the pre-set proportion of the given number that have sequential logical addresses by writing their data into the first designated block with sequential physical addresses, and

responding to host commands to write a number of units of data having sequential logical addresses that is equal to or in excess of the pre-set proportion of said given number by writing their data into a block other than the first designated block.

3. The method of claim 2, which additionally comprises, prior to writing data of the plurality of successive host commands:

determining whether or not the successive host commands individually include a number of units of data having sequential logical addresses less than the pre-set proportion of said given number.

4. The method of claim 2, wherein the non-volatile memory cells are organized into multiple sub-arrays and said blocks of memory cells include memory cells of two or more of the sub-arrays.

26. The method of claim 2, wherein the pre-set proportion is set within a range of 25-75 percent of said given number.

27. The method of claim 3, wherein the pre-set proportion is set within a range of 25-75 percent of said given number.

28. A method of writing data into a non-volatile memory system of a type having blocks of memory cells that are simultaneously erasable and which individually store a given number of host units of data, comprising:

designating at least a first one of the blocks to store a number of units of data having sequential logical addresses that is less than a pre-set fraction of said given number,

thereafter responding to a plurality of successive host commands to individually write units of data into the memory system by determining whether a number of the units of data with sequential logical addresses is less than the pre-set fraction, and, if so, by writing the data into the first dedicated block, and

responding to host commands to write units of data having a number of sequential logical addresses that is equal to or in excess of the pre-set fraction of said given number by writing the data into a block other than the first dedicated block.

29. The method of claim 28, wherein the non-volatile memory cells are organized into multiple sub-arrays and said blocks of memory cells include memory cells of two or more of the sub-arrays.

30. The method of claim 28, wherein the fraction is set within a range of 25-75 percent of said given number.

31. A method of operating a non-volatile memory system in response to commands received from a host to individually write logically addressed units of data therein, the memory system having memory cells grouped into blocks of memory cells that are simultaneously erasable and which individually store a given number of units of data at individual physical addresses, the logical addresses of received units of data being mapped within the memory system into corresponding physical addresses where the received units of data are stored, comprising:

allocating a first one of the blocks to store units of data having a number of sequential logical addresses less than a pre-determined fraction of said given number,

allocating a second one of the blocks to store units of data having a number of sequential logical addresses equal to or in excess of the fraction of said given number,

in response to receipt of a command to write data having a number of sequential logical addresses less than said fraction, determining whether the first block has sufficient erased capacity to store the received data and, if so, writing the received data into sequential physical addresses of the first block, and

in response to receipt of a command to write data having a number of sequential logical addresses equal to or in excess of said fraction, determining whether the second block has erased capacity to store the data and, if so, writing the data into sequential physical addresses of the second block.

32. The method of claim 31, additionally comprising:

in response to receipt of the command to write data having a number of sequential logical addresses less than said fraction, if the first block does not have sufficient erased capacity to store the received data, allocating a third one of the blocks to store units of data having a number of sequential logical addresses less than a fraction of said given number and then writing the received data into sequential physical addresses of the third block, and

in response to receipt of the command to write data having a number of sequential logical addresses equal to or in excess of said fraction, if the second block does not have sufficient erased capacity to store the received data, allocating a fourth one of the blocks to store units of data having a number of sequential logical addresses equal to or in excess of the fraction of said given number and then writing the received data into sequential physical addresses of the fourth block.

33. The method of claim 32, wherein the fraction is set to be within a range of 25-75 percent of said given number.

34. The method of claim 31, wherein the non-volatile memory cells are organized into multiple sub-arrays and said blocks of memory cells include memory cells of two or more of the sub-arrays.

35. The method of claim 31, wherein the fraction is set to be within a range of 25-75 percent of said given number.

36. A method of writing data into a non-volatile memory system of a type having blocks of memory cells that are simultaneously erasable and which individually store a given number of host units of data, comprising:

designating at least a first one of the blocks to store a number of units of data received by the memory system with individual ones of multiple write commands that have sequential logical addresses less than a pre-set fraction of said given number,

responding to the receipt of multiple commands by the memory system to individually write one or more units of data thereinto by, for individual commands,

(a) determining whether the command specifies the writing of a number of units of data having sequential logical addresses that is less than the pre-set fraction, and

(b) determining whether the first block has enough erased capacity to store the number of units of data provided with the command, wherein

when both of conditions (a) and (b) above are determined to exist, thereafter writing the units of data into the first block, but

when either one of conditions (a) or (b) above is determined not to exist, writing the units of data into one of the blocks other than the first block.

37. The method of claim 36, wherein the pre-set fraction is within a range of 25-75 percent of said given number.

38. The method of claim 36, additionally comprising:

designating at least a second one of the blocks to store a number of units of data received by the memory system with individual ones of multiple write commands that have sequential logical addresses equal to or greater than the pre-set fraction, and

responding to the receipt of multiple commands by the memory system to individually write one or more units of data thereinto by additionally, for individual commands,

(c) determining whether the command specifies the writing of a number of units of data greater than the given number, wherein

when neither of the conditions (a) nor (c) above exist, writing the units of data into the second block, without regard to whether condition (b) exists or not, but

when the condition (c) above is determined to exist, writing the units of data into one of the blocks other than the first or second blocks.

39. The method of claim 38, wherein the pre-set fraction is set to be within a range of 25-75 percent of said given number.

40. In a non-volatile memory system having memory cells grouped into blocks of memory cells that are simultaneously erasable and which individually store a given number of units of data at individual physical addresses, the logical addresses of received units of data being mapped within the memory system into corresponding physical addresses where the received units of data are stored, a method of operation in response to received commands to individually write logically addressed units of data therein, comprising:

designating a first one of the blocks to store units of data having a number of sequential logical addresses that is less than a pre-determined fraction of said given number,

designating a second one of the blocks to store units of data having a number of sequential logical addresses that is greater than the fraction of said given number,

providing at least another one of the blocks that is fully erased, and

in response to receipt of a command to write data into the memory system, identifying the number of units of the data that have sequential logical addresses, determine whether the number of such units with sequential logical addresses are less than the fraction, and, if so,

writing the data to the first of the blocks, but if the amount of data is not less than the fraction, then

writing the data to the second of the blocks if there is sufficient capacity therein, but if there is not sufficient capacity in the second of the blocks, writing the data to the fully erased block.

41. The method of claim 40, wherein the pre-predetermined fraction is set to be within a range of 25-75 percent of said given number.

IX. EVIDENCE APPENDIX

None

X. RELATED PROCEEDINGS APPENDIX

None